

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS

TEMA:
DESARROLLO DE UN DISPOSITIVO DE RASTREO PARA PERSONAS
ADULTAS MAYORES EN UNA RED IOT Y CLOUD.

AUTORES:
JUAN FRANCISCO DÍAZ SALAZAR
JIMMY FABRICIO ROJAS MASABANDA

TUTOR:
LUIS GERMÁN OÑATE CADENA

Quito, mayo del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Juan Francisco Díaz Salazar y Jimmy Fabricio Rojas Masabanda, con documento de identificación N° 172521608-7 y N° 050333591-1, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores del trabajo de titulación intitulado: **DESARROLLO DE UN DISPOSITIVO DE RASTREO PARA PERSONAS ADULTAS MAYORES EN UNA RED IOT Y CLOUD.**, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, mayo 2021



Juan Francisco Díaz Salazar
172521608-7



Jimmy Fabricio Rojas Masabanda
050333591-1

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico DESARROLLO DE UN DISPOSITIVO DE RASTREO PARA PERSONAS ADULTAS MAYORES EN UNA RED IOT Y CLOUD., realizado por Juan Francisco Díaz Salazar y Jimmy Fabricio Rojas Masabanda, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, mayo 2021



Luis Germán Oñate Cadena

CI: 1712157401

DEDICATORIA

Dedico mi trabajo realizado con mucho cariño y de corazón a mis Padres Segundo Rojas y Gladys Masabanda por su enorme esfuerzo y sacrificio, por el apoyo incondicional brindado en cada momento difícil, por ayudarme a construir mi vida profesional y por educarme para ser una persona de bien, los quiero mucho.

De igual forma a Dios por guiarme y darme fuerzas cada día para seguir adelante y ayudarme a cumplir mis metas. Y a mis hermanas por el afecto, compañía y por los grandes momentos compartidos que sin esperar nada a cambio siempre me apoyan, gracias por las alegrías y tristezas que hemos pasado.

Jimmy Rojas

Este trabajo se lo dedico a mis padres Segundo Juan Díaz y Ana María Salazar quienes nunca dejaron de creer en mí, por su apoyo incondicional y todos los sacrificios que hicieron para darme la oportunidad de formarme tanto personal como profesionalmente, a mi familia y en especial a Dios que hizo todo esto posible.

Y a mis amigos, especialmente a mi compañero de tesis Jimmy por estar ahí en los buenos y malos momentos, y a Tania por sus palabras de aliento, que me impulsaron en todo mi trayecto universitario.

Juan Díaz

AGRADECIMIENTO

Agradezco a Dios por regalarme la vida y disfrutar de una familia unida, por guiarme todo este tiempo evitando todo tipo de peligros y aprender a disfrutar la vida.

Muy agradecido con mis padres, mis hermanas, mis amigos y con la Universidad Politécnica Salesiana quienes han formado parte de mi vida y me formaron profesionalmente cumpliendo su misión de formar honrados ciudadanos y buenos cristianos.

Al Ingeniero Luis Germán Oñate tutor en este proyecto, por su valioso apoyo y valiosa experiencia a quien le tengo un gran respeto y admiración, nos guio y ayudo a culminar nuestro trabajo

Gracias a mis docentes de la carrera de Ingeniería Electrónica por compartir su amplio conocimiento a lo largo de mi formación profesional.

Jimmy Rojas

Agradezco a Dios por darme la oportunidad de rodearme de una familia amorosa y buenas amistades, por brindarme la sabiduría y coraje de afrontar todos los problemas hasta ahora y espero seguir contando con sus bendiciones durante toda mi vida.

Infinitamente agradecido con mis padres, hermanos y amigos, con esta institución que me acogió durante todo este tiempo, que me instruyó como profesional y como persona, y todo el personal docente que tuvo la amabilidad de transmitir sus conocimientos en mi formación, especialmente al Ingeniero Luis German Oñate con quien comparto los mismos sentimientos de agradecimiento que mi compañero.

Juan Díaz

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR	i
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE GENERAL	v
ÍNDICE DE FIGURAS	vii
ÍNDICE DE TABLAS	viii
RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN	xi
CAPÍTULO 1	1
ANTECEDENTES	1
1.1. Planteamiento del problema	1
1.2. Justificación	2
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
1.4. Metodología	3
1.4.1. Metodología Descriptiva	3
1.4.2. Metodología Analítica	3
1.4.3. Metodología Experimental	3
CAPÍTULO 2	4
FUNDAMENTACIÓN TEÓRICA	4
2.1. Servicio General de Paquetes Vía Radio	4
2.2. Internet de las cosas (IoT)	4
2.3. Cloud	5

2.3.1 Firebase	5
2.3.2 Android Studio.....	6
2.4 Tablas de Decisión	7
2.5. Arduino Nano	7
2.5.1 Módulo GSM, GPRS SIM800L.....	8
2.5.2 Módulo GPS GY-GPS6MV2	10
CAPÍTULO 3	12
DISEÑO DEL DISPOSITIVO	12
3.1 Diagrama de bloques del dispositivo.....	12
3.2 Diagramas de Flujo	14
3.3 Diagrama Esquemático.....	21
3.4 Aplicación para la prevención de desapariciones de adultos mayores.....	23
3.5 Geolocalización de la persona de la tercera edad.....	30
3.6 Resultados Obtenidos.	31
CAPÍTULO 4	37
ANÁLISIS DE COSTOS	37
4.1 Costos de Materiales y Módulos	37
4.2. Indicadores para la factibilidad del Proyecto	38
4.2.1. Costo Fijo.....	38
4.2.2. Costo Variable.	39
4.2.3. Punto de Equilibrio	39
CAPÍTULO 5	40
CONCLUSIONES.....	40
RECOMENDACIONES	41
REFERENCIAS BIBLIOGRÁFICAS	42
ANEXOS	44

ÍNDICE DE FIGURAS

Figura 2.1 Android Studio.	6
Figura 2.2 Arduino Nano.	8
Figura 2.3 Módulo SIM800L	9
Figura 2.4 Módulo GPS GY-GPS6MV2.	11
Figura 3.1 Diagrama de bloques que integra las partes del proyecto de rastreo GPF.	12
Figura 3.2 Diagrama de bloques para la comunicación entre el dispositivo y la nube.	13
Figura 3.3 Diagrama de flujo para obtener datos mediante Arduino.	14
Figura 3.4 Diagrama de flujo de la recepción de datos obtenidos del módulo GPS..	15
Figura 3.5 Diagrama de Flujo para la transmisión de datos obtenidos por el módulo GPS.	16
Figura 3.6 Diagrama de flujo para la interfaz inicial del uso de la aplicación (Pantalla 1).	17
Figura 3.7 Diagrama de flujo para la interfaz inicial de la aplicación (Pantalla 1)....	18
Figura 3.8 Diagrama de flujo para la interfaz inicial del uso de la aplicación (Pantalla 2).	19
Figura 3.9 Conexiones de los módulos con la tarjeta de Arduino nano.....	21
Figura 3.10 Diagrama Esquemático, diseñado en Proteus.	22
Figura 3.11 Estructura del Equipo.	23
Figura 3.12 Interfaces de la aplicación desarrollada en Android Studio.	24
Figura 3.13 Estado de la persona: Dentro de casa.	25
Figura 3.14 Estado de la persona: En peligro.	26
Figura 3.15 Estado de la persona: Actualizando.	27
Figura 3.16 Vinculación con Servicios de rutas Google Maps.	28
Figura 3.17 Vinculación con Servicios de lugares de Google Maps.	29
Figura 3.18 Finalización del ciclo de monitoreo.....	30
Figura 3.19 Análisis de precisión y exactitud de la posición del dispositivo.	34
Figura 3.20 Prueba de Campo.....	35
Figura 3.21 Área aceptable.	36

ÍNDICE DE TABLAS

Tabla 2.1 Matriz de doble entrada.....	7
Tabla 2.2 Consumo del módulo SIM800L según su modo de operación y frecuencia.	10
Tabla 3.1 Mediciones de exactitud y precisión del dispositivo.	32
Tabla 4.1 Materiales y Módulos.	37
Tabla 4.2 Precio Total del Proyecto.....	38

RESUMEN

La carencia de soluciones ante la problemática de las desapariciones de personas adultas mayores en nuestro país, se han vuelto un inconveniente en el proceso de localización de la persona en riesgo por parte de las autoridades, tomando en cuenta las circunstancias de salud corporal como mental de dichas personas, hacen que las probabilidades que una de estas personas entre en situación de vulnerabilidad aumenten, para lo cual en este presente proyecto se propone el desarrollo de un dispositivo de rastreo IOT y Cloud que permita prevenir que el adulto mayor incurra en una desaparición, el diseño propuesto consta de un conjunto de módulos GPS y GSM en la plataforma de Arduino, con la ayuda del software Android Studio en vinculación con Firebase y las plataformas de servicios de Google Play asociadas, haciendo posible el uso de un aplicativo móvil GranPa Finder(GPF) que permite observar los movimiento de la persona en riesgo en tiempo real, además de trazar rutas hacia la última posición conocida en el ciclo de monitoreo, así como, brindar lugares de referencia con alta precisión, además de alertas tempranas y recurrentes para que los familiares y las personas a cargo de su cuidado puedan tener la seguridad de que el adulto mayor no corra peligro.

ABSTRACT

The lack of solutions to the problem of disappearances of elderly people in our country, have become a drawback in the process of locating the person at risk by the authorities, taking into account the circumstances of bodily and mental health of these people, make the chances that one of these people enter into a situation of vulnerability increase, for which this project proposes the development of an IOT and Cloud tracking device that prevents the elderly incur a disappearance, The proposed design consists of a set of GPS and GSM modules on the Arduino platform, with the help of Android Studio software in conjunction with Firebase and associated Google Play service platforms, making possible the use of a mobile application GranPa Finder (GPF) that allows observing the movement of the person at risk in real time, It can also trace routes to the last known position in the monitoring cycle, as well as provide highly accurate reference locations and early and recurrent alerts so that family members and caregivers can be assured that the elderly person is not at risk.

INTRODUCCIÓN

La situación actual del país con respecto a la prevención de desapariciones deja mucho que desear, por lo general el sistema funciona a partir de la desaparición, con protocolos que han sido utilizados por casi 10 años, tratando de ubicar a la persona en riesgo con la incertidumbre de poder obtener las pistas necesarias para localizarla en el menor tiempo posible para el alivio de los familiares y personas encargadas de su cuidado. (LEXIS, 2013)

Las personas adultas mayores con el paso del tiempo tienden a encontrarse con situaciones que las ponen en un estado vulnerable, sea por el decaimiento de sus sentidos básicos, enfermedades o situaciones psicológicas que afectan a la posibilidad de enfrentar una vejez tranquila. Bajo estas condiciones las personas son más propensas a tomar decisiones que ponen en riesgo su integridad, esto lastimosamente se sale del control de su familia o de las personas que se encargan de su cuidado.

Se desarrolló un dispositivo de rastreo IOT y Cloud para la prevención de que estas situaciones de riesgo ocurran, acudiendo a la tecnología como herramienta de monitoreo en tiempo real del adulto mayor, definiendo un área aceptable que pueda ser determinada por el usuario mediante la plataforma de Arduino nano y el software Arduino (IDE) en conjunto con los módulos GSM SIM800 y GPS NEO 6m para una red IOT y Cloud para monitorear la posición de la persona en cuestión en tiempo real y subir los datos a la base de datos Firebase, estos datos son utilizados dentro de un aplicativo GranPa Finder (GPF) desarrollado en la plataforma de Android Studio y usando los servicios de Google Maps para graficar la trayectoria recorrida por el adulto mayor y la posibilidad de generar una ruta hacia el destino, además se implementó medidas de alerta usando la red móvil para enviar mensajes a los dispositivos móviles de los encargados del cuidado de los adultos mayores.

Este trabajo tiene cinco capítulos se describen a continuación:

En el Capítulo 1: Planteamiento del problema, Justificación, Objetivo general y específicos y la metodología.

En el Capítulo 2: Fundamentación teórica e información sobre IOT y Cloud, además de los parámetros de los componentes para el desarrollo del dispositivo.

En el Capítulo 3: Se detalla el diseño e implementación del hardware y software del aplicativo y el envío del mensaje de alerta.

En el Capítulo 4: Se analizó los costos para determinar la aceptación del equipo.

En el Capítulo 5: Se incluyen las conclusiones y recomendaciones.

Además, se tiene los siguientes anexos: Código de programación en Arduino y código de programación del GPD en Android Studio.

CAPÍTULO 1

ANTECEDENTES

1.1. Planteamiento del problema

En una ciudad en crecimiento como es la ciudad de Quito, caracterizada por rápidos y complejos cambios tanto en su situación demográfica, como territorial abarca ciertos problemas que persisten a lo largo de los años , y que las autoridades no han encontrado una salida viable en lo que respecta a la desaparición de personas, los esfuerzos de los equipos de investigación se hacen más complejos a medida que pasa el tiempo debido a que es incierta la situación de la persona involucrada en este evento.

Por esto, cada vez tendrán que lidiar con estos eventos de una forma más innovadora, hay que tomar en cuenta que, desde hace mucho tiempo que no se cambia el protocolo para esta clase de problemas de carácter social entonces, aliarse con las nuevas tecnologías será de vital importancia para salvaguardar la vida y la integridad de las personas.

En la actualidad no existe ninguna tecnología en la ciudad de Quito que preste ayuda a las personas propensas a desapariciones de forma inmediata a partir del suceso.

El rastreo preventivo de una desaparición sería de mucha ayuda para las autoridades para evadir este hecho, y en el peor de los casos brindar información para que la persona involucrada sea asistida en el menor tiempo posible.

Otro factor a considerar es que las personas responsables del adulto mayor no pueden estar todo el tiempo pendiente de su ubicación por diferentes razones, además la mayoría de los adultos mayores no manejan de manera adecuada la tecnología actual haciendo que la comunicación constante con la persona en cuestión sea más dificultosa.

1.2. Justificación

La desaparición de personas de la tercera edad es una condición donde el tiempo de localización es esencial para salvaguardar la integridad de la víctima, ya que al no contar con la vigilancia o comunicación las 24 horas con la persona en cuestión, el saber el lugar exacto hacia donde se dirige es muy difícil de descifrar y tomando en cuenta que dicha persona es muy propensa a ser manipulada sin saberlo, el riesgo que toma aumenta con el pasar del tiempo. El desarrollo de este dispositivo permitirá a sus familiares más cercanos o personas de entera confianza del desaparecido saber la localización exacta en tiempo real emitiendo mensajes con su última ubicación disponible, siendo estos datos vitales en las primeras etapas de la desaparición y previniendo que la situación se vuelva más compleja.

1.3. Objetivos

1.3.1. Objetivo General

- Desarrollar un dispositivo para red IOT y Cloud que determine la ubicación en tiempo real de una persona adulta mayor.

1.3.2. Objetivos Específicos

- Implementar un dispositivo rastreador de la ubicación de una persona mediante posicionamiento global para que este envíe alertas mediante una red IOT a los familiares que cuidan a la persona de la tercera edad.
- Implementar en Cloud un sistema de registro para que se determine el tiempo que permaneció en un lugar determinado.
- Verificar el correcto funcionamiento del dispositivo en tiempo real mediante pruebas de campo.
- Analizar el costo del dispositivo, para que se determine la factibilidad de implementación del dispositivo para uso de personas adultas mayores.

1.4. Metodología

1.4.1. Metodología Descriptiva

Se utilizará esta metodología para establecer las características de los módulos GPS y GSM estableciendo la relación entre ellos.

1.4.2. Metodología Analítica

Se efectuará un análisis de la factibilidad del proyecto en base a los costos del servicio propuesto, con las medidas actuales en casos de desaparición.

1.4.3. Metodología Experimental

Mediante la experimentación se comprobará el funcionamiento del sistema en el cual se monitorea los movimientos en tiempo real de un adulto mayor.

CAPÍTULO 2

FUNDAMENTACIÓN TEÓRICA

En este capítulo se describe resumidamente los aspectos técnicos y principales fundamentos de los elementos del dispositivo, además del software base en el cual se desarrolla la interfaz de usuario y monitoreo.

Las tecnologías que están envueltas en nuestro dispositivo son:

- Servicio General de Paquetes Vía Radio (GPRS): Nos ayuda a la transmisión de datos mediante la red GSM.
- Sistema de Posicionamiento Global (GPS): Nos permite determinar coordenadas geográficas (latitud, longitud) del dispositivo en todo el planeta.
- Internet de las Cosas (IoT) y Cloud: Nos permite almacenar, visualizar y modificar en una base de datos en tiempo real las variables a utilizar.
- Arduino: Nos ayuda al desarrollo del dispositivo con sus diferentes componentes modulares de fácil manipulación.

2.1. Servicio General de Paquetes Vía Radio

El Servicio General de Paquetes Vía Radio (GPRS) es un sistema de telefonía móvil de segunda generación también conocida como 2.5G que proporciona servicio de datos de alta velocidad y una versión mejorada de la red GSM, fue una de las primeras infraestructuras de la red móvil que permitió conectarse a la red de Internet habilitando la navegación por la web desde un teléfono celular a través de redes de datos. Entre sus beneficios tenemos servicio de MMS (Servicios de mensajes multimedia), (SMS) Servicios de mensajes cortos, PTP (Servicio punto a punto) y PTMP (Servicio punto a multipunto), que contaba en la tecnología 2G, pero utiliza señal GSM; La velocidad de transferencia máxima teórica va entre 52 y 171.2 Kbps y el pago del servicio es por la cantidad de información transmitida y no por el tiempo de conexión. (Wevar, 2005)

2.2. Internet de las cosas (IoT)

En el 2020 se vio un crecimiento global en la adquisición de tecnología, la digitalización

llego a ser soporte desde la educación, el trabajo, la telemedicina y prensa digital, por lo que la comodidad dentro de los hogares jugó un papel fundamental. La tecnología IoT está en constante desarrollo y cada día es más accesible a que más personas hagan uso de ella, a inicios el IoT estaba pensada para conectar dispositivos electrónicos a una red Wi-Fi como televisores, lavadoras, entre otras, sin embargo, con la aparición de redes de quinta generación como el 5G se da un salto a modernizar la capacidad del tradicional IoT y ya no se hablan de dispositivos conectados a la red sino de cosas conectadas a la red. El despliegue de las redes móviles de quinta generación (5G) implica la ampliación inminente de los beneficios de la vida digital, que promete mayor diversidad y penetración de los servicios virtuales que lo logrado con las actuales redes. (Luna, 2020)

2.3. Cloud

Cloud consiste en la oportunidad de brindar servicios a través de internet, esta tecnología busca almacenar todos nuestros datos personales como: archivos, imágenes, documentos, etc. y hacerlos disponibles inmediatamente sin depender de la capacidad de memoria de un dispositivo físico siempre y cuando tengamos acceso a una red. (Chango Quinapanta, 2013)

2.3.1 Firebase

Firebase es una plataforma de Google cuya función principal es desarrollar y facilitar la creación de apps de elevada calidad de una forma rápida. Esta plataforma está subida en la nube y está disponible en iOS, Android y Web lo que hace muy versátil y adaptable a la medida de las necesidades del usuario (Cardona, 2019). Sus principales características se pueden encontrar en el siguiente enlace: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>

Firebase mediante archivos de formato JSON permite la sincronización con varios desarrolladores de aplicaciones móviles ya que son menos pesados que los convencionales archivos XML o SQL, además permanecen actualizados sin que el

usuario realice ninguna configuración adicional, si no hay una conexión por parte del usuario los datos siguen disponibles hasta que vuelva a conectarse a la red y se sincronicen los datos. (López, 2020)

2.3.2 Android Studio

Android Studio es un entorno de desarrollo (IDE) oficial para el desarrollo de apps para Android y basado en IntelliJ IDEA. Posee un potente editor de códigos que aumentan la productividad de las apps (Developers Android, s.f.), algunas características destacables las encontramos en el siguiente enlace: <https://developer.android.com/studio/intro?hl=es-419>

Figura 2.1 Android Studio.



Autor: (Pérez, 2017)

En la figura 2.1 se puede apreciar la plataforma de software en el cual está desarrollado el proyecto, Android Studio tiene distintos componentes que ayudan a la tarea de desarrollar aplicaciones como la construcción basada en Gradle que es una herramienta que permite la automatización de compilación de código abierto, además de creación de variantes y múltiples archivos APK, como también plantillas que simplifican el proceso de inicio de la aplicación.

2.4 Tablas de Decisión

Tabla 2.1 Matriz de doble entrada.

Características Tarjetas de desarrollo	Compatibilidad Módulo SIM800L	Compatibilidad Módulo GPS GY-NEO 6M	Fácil de adquirir en Ecuador	Fácil de aprender a programar	Precio			Consumo		Tamaño		
					Alto	Medio	Bajo	Mayor a 2A	Menor a 2A	Grande	Mediano	Pequeño
Raspberry Pi	x	x	x	x	x			x		x		
Arduino Nano	x	x	x	x		x			x			x
PIC 16F877A	x	x	x				x		x			x
NodeMCU	x	x	x	x		x			x		x	
STM32	x	x	x			x			x			x

Elaborado por: Los Autores.

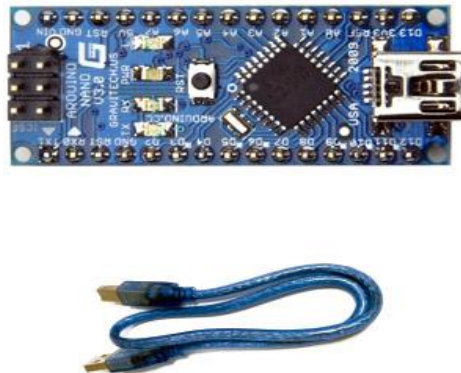
En la tabla 2.1 se tomó en cuenta las características de cada uno de los dispositivos que tentativamente se preveían utilizar y que tan accesible resulta para el proyecto, se destacó el Arduino Nano por su versatilidad, dimensiones, precio, por la gran cantidad de módulos accesibles tanto en precio como configuración, el consumo energético y la facilidad del lenguaje de programación que facilite la viabilidad de las comunicaciones inalámbricas con los servidores de IoT y Cloud.

2.5. Arduino Nano

El módulo Arduino Nano es una placa de desarrollo basada en el microcontrolador ATMEGA328P(SMD), programable, diseñada para proyectos electrónicos y de robótica que nos permite la conexión de varios módulos compatibles con su configuración de modo muy versátil y de bajos consumos de energía. (Master, 2020)

Entre algunas características técnicas importantes tenemos que cuenta con un microcontrolador ATmega328, voltaje de operación de 5V y 14 pines de entrada y salida digital. Para revisar mas detalles de las especificaciones técnicas visite el siguiente enlace: <https://store.arduino.cc/usa/arduino-nano>

Figura 2.2 Arduino Nano.



Autor: (Master, 2020)

En la Figura 2.2 se puede apreciar la placa de desarrollo con sus periféricos y cable de comunicación USB tipo mini B también conocido como mini B.

2.5.1 Módulo GSM, GPRS SIM800L

Módulo celular a pequeña escala y bajo costo que permite enviar y recibir, datos, SMS y llamadas. Sus características se detallan de mejor forma en el siguiente enlace: <https://www.naylampmechatronics.com/inalambrico/115-modulo-gsm-sim800l-2g.html>

Figura 2.3 Módulo SIM800L



Autor: (Electronics, Naylamp, 2020)

Para el correcto funcionamiento del módulo es contar con una fuente que suministre la suficiente energía para su funcionamiento, el consumo de corriente del módulo puede llegar hasta los 2A durante la transmisión y registro en la red, en la figura 2.3 se puede observar una representación del módulo GSM. El módulo no incorpora un regulador de tensión por lo que es necesario una fuente externa que suministre esos 4V; Dependiendo del estado del módulo el consumo de energía cambia, estos parámetros se pueden observar resumidas a continuación y para un detalle completo revisar el datasheet del módulo en el siguiente enlace: https://img.filipeflop.com/files/download/Datasheet_SIM800L.pdf

Tabla 2.2 Consumo del módulo SIM800L según su modo de operación y frecuencia.

Modo	Frecuencia	Consumo
Power down		60 uA
Sleep mode		1.11 mA
Llamada de Voz (Voice call)	GSM850	199.48 mA
	EGSM900	216.12 mA
	DCS1800	146.48 mA
	PCS1900	131.77 mA
GPRS	GSM850	453.57 mA
	EGSM900	344.74 mA
	DCS1800	216.20 mA
	PCS1900	274.04 mA

Elaborado por: Los Autores.

En la tabla 2.2 se hace la relación de consumo de energía, con las diferentes configuraciones de frecuencia tanto para llamadas de voz como servicio GPRS.

2.5.2 Módulo GPS GY-GPS6MV2

El Módulo GPS GY-GPS6MV2 es un producto de la serie U-Blox NEO 6M, equipado con una EEPROM con configuración de fábrica, indicadores de funcionamiento LED y una antena cerámica que nos permite mediante la interfaz serial de un microcontrolador, observar los datos de posición global en tiempo real, con configuraciones personalizadas. (Electronics, Naylamp, 2016), para un mayor detalle de las características ingresar al siguiente enlace: https://www.naylampmechatronics.com/blog/18_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html

Figura 2.4 Módulo GPS GY-GPS6MV2.



Autor: (robótica, s.f.)

En la figura 2.4 se observa el módulo GPS conectado a su antena cerámica y los pines para la comunicación serial RX, TX y los pines de alimentación VCC y GND.

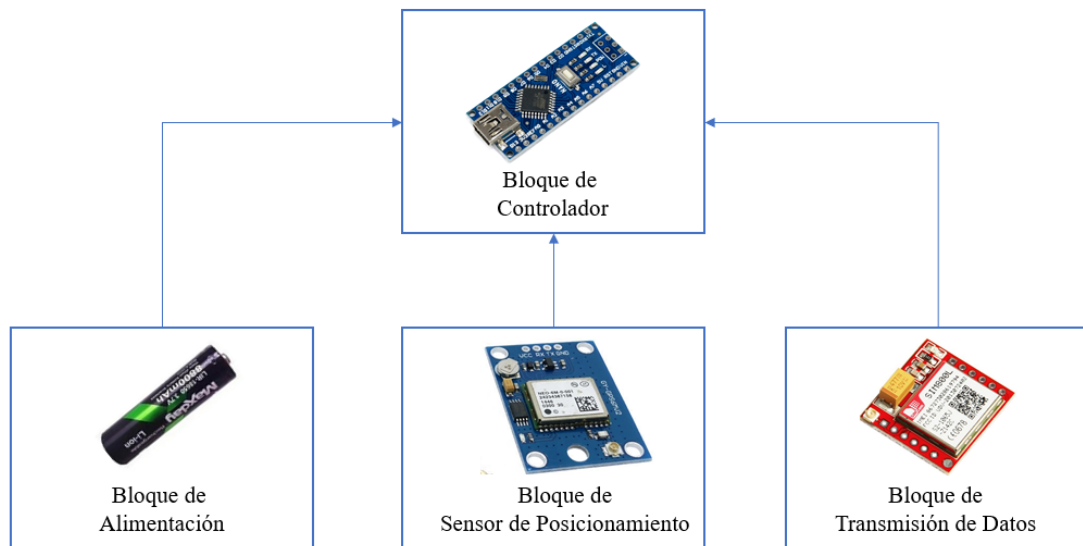
CAPÍTULO 3

DISEÑO DEL DISPOSITIVO

En este capítulo se va a tratar todo lo que respecta al diseño de hardware y software del dispositivo de rastreo, así como las diferentes configuraciones, diagramas de bloque y flujo de cada uno de los elementos utilizados en el mismo para su desarrollo.

3.1 Diagrama de bloques del dispositivo.

Figura 3.1 Diagrama de bloques que integra las partes del proyecto de rastreo GPF.



Elaborado por: Juan Diaz, Jimmy Rojas

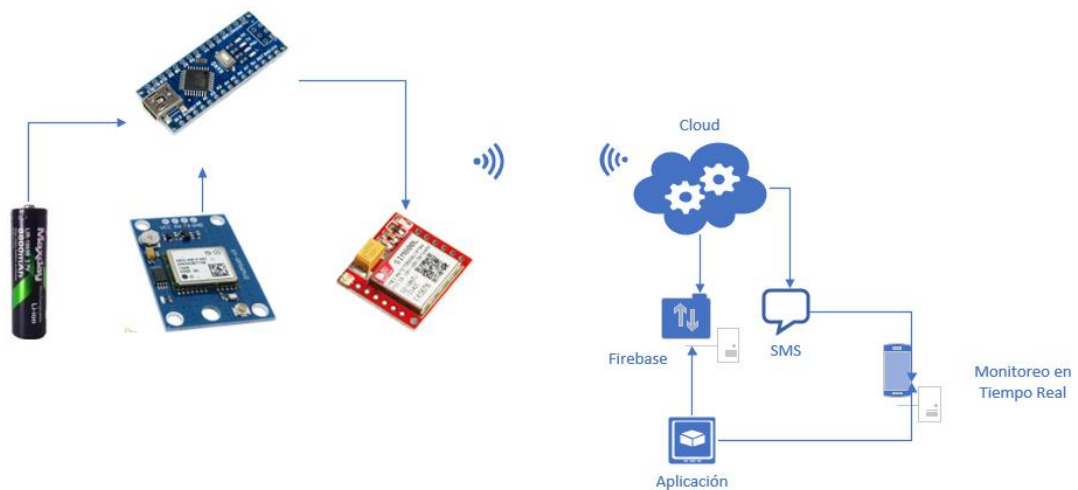
En la figura 3.1 se representa los elementos utilizados para el diseño, como el bloque de alimentación que se encarga de suministrar la energía a los diferentes componentes, el cual se adaptó al voltaje de trabajo requerido de 5V.

Para el bloque del controlador se utilizó una Arduino nano, que es el módulo central encargado de llevar a cabo el procesamiento de los datos y el encargado de controlar el trabajo conjunto de los demás módulos.

El bloque del sensor GPS es el módulo que se encarga de obtener los datos de posición en tiempo real y transmitirlos al módulo de Arduino para su procesamiento y análisis de esos datos para su posterior envío hacia nuestro servidor.

El bloque de transmisión es el encargado de realizar la comunicación inalámbrica, se encarga de transmitir los datos mediante GPRS y subirlos a la nube.

Figura 3.2 Diagrama de bloques para la comunicación entre el dispositivo y la nube.

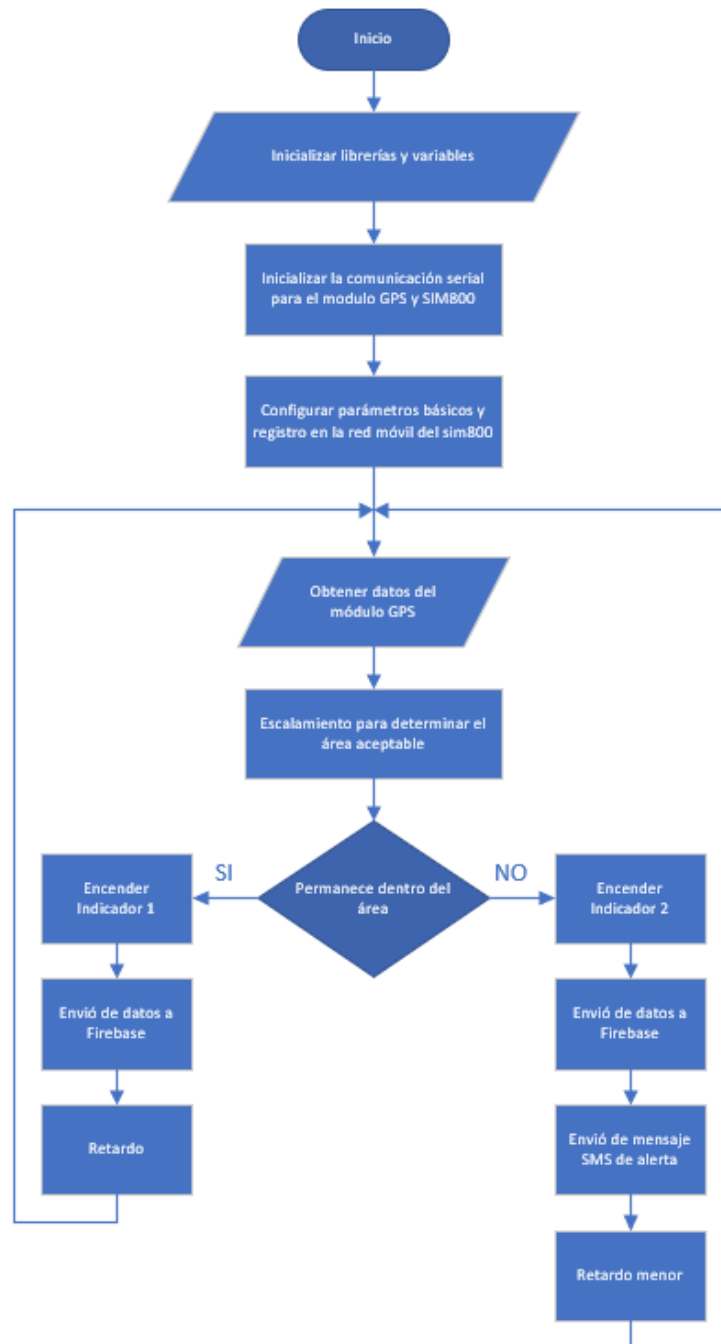


Elaborado por: Juan Diaz, Jimmy Rojas

En la figura 3.2 se puede observar los elementos que hacen parte del envío de datos a la plataforma de Cloud mediante el módulo SIM800L, esto hará posible subir la información de longitud y latitud a una base de datos en tiempo real y también permitirá el monitoreo mediante un móvil inteligente que tenga conexión a internet.

3.2 Diagramas de Flujo

Figura 3.3 Diagrama de flujo para obtener datos mediante Arduino.

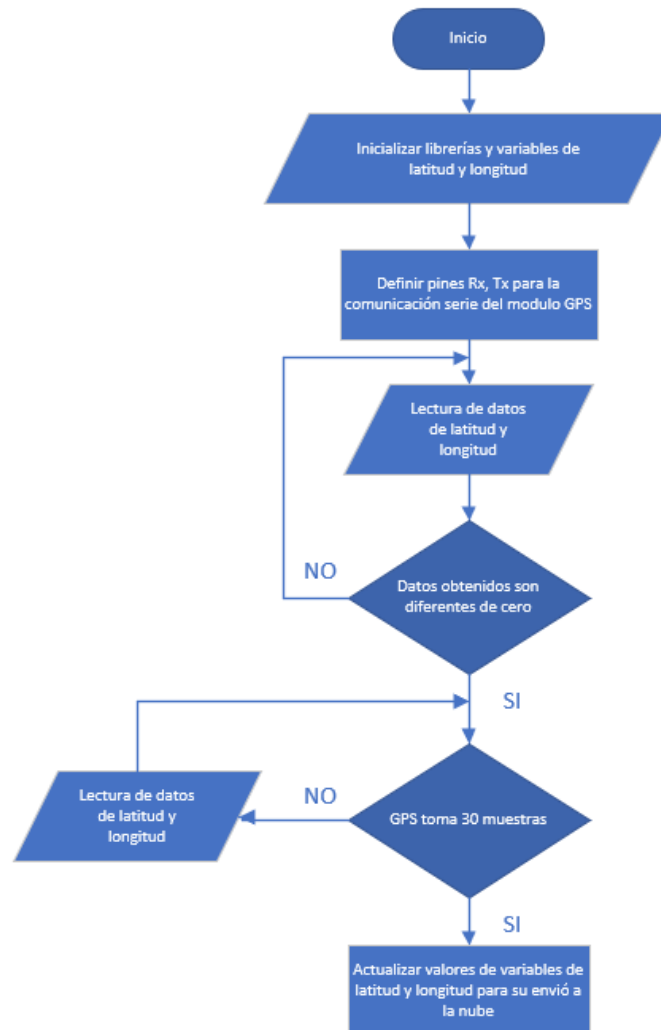


Elaborado por: Juan Diaz, Jimmy Rojas

El algoritmo desarrollado para la recepción y transmisión de datos como se observar en la figura 3.3, inicia con declarar las variables de latitud, longitud y las librerías necesarias, se inicializa la comunicación serial entre el módulo GPS y el SIM800 para

establecer comunicación, se verifica que el módulo SIM800 establezca comunicación serial con el Arduino y que se registre en la red móvil de la operadora mediante comandos AT, se obtiene los datos de latitud y longitud que proporciona el módulo GPS y se realiza el escalamiento para determinar el área aceptable mediante la fórmula de distancia entre dos puntos, evaluamos el valor del escalamiento para determinar si sigue dentro del área aceptable y enviar su posición en caso de ser verdadero cada cierto tiempo, caso contrario se envía la posición en periodos más cortos de tiempo y un mensajes de alerta informando que el sujeto ha salido del área establecida y se tomen las medidas respectivas.

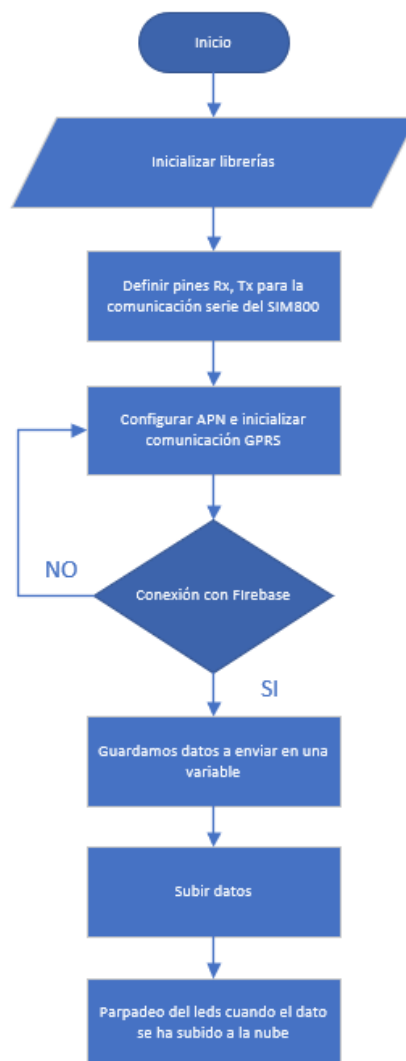
Figura 3.4 Diagrama de flujo de la recepción de datos obtenidos del módulo GPS.



Elaborado por: Juan Diaz, Jimmy Rojas

En la figura 3.4 se muestra el subproceso para la obtención de los datos de posicionamiento, se inicializa las variables de latitud y longitud de nuestra área y la librería donde se asigna los pines utilizados para la comunicación, después se valida que el GPS reciba datos validos antes de realizar el escalamiento para determinar el área aceptable, verificamos si los datos leídos por el GPS son distintos de cero, caso contrario volvemos a validar y comprobar si el GPS está recibiendo datos verdaderos, si son distintos de cero guardamos esos datos y procedemos a tomar 30 lecturas de la posición para su escalonamiento y posteriormente ser enviados.

Figura 3.5 Diagrama de Flujo para la transmisión de datos obtenidos por el módulo GPS.



Elaborado por: Juan Diaz, Jimmy Rojas

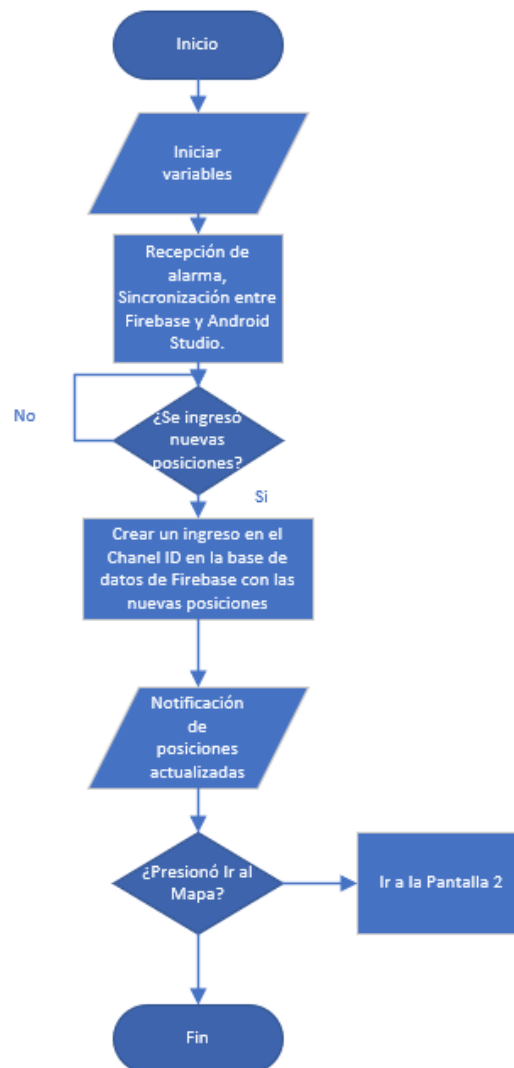
La figura 3.5 muestra el algoritmo para el envío de datos con el módulo SIM800L, se inicializa la librería y se especifica los pines para la comunicación entre el módulo y el controlador, se configura el APN, usuario y contraseña de la operadora móvil para acceder a internet, una vez configurado los parámetros anteriores se activa el modo de conexión GPRS y el servicio HTTPS del SIM800 mediante comandos AT, se verifica si se establece conexión con Firebase para subir los datos de latitud, longitud, hora, minutos y segundos obtenidos del dispositivo, cuando termina de subir los datos a la nube se muestra un parpadeo de leds indicando que el dato se terminó de subir, caso contrario se realiza nuevamente las peticiones para la conexión GPRS y servicio HTTPS.

Figura 3.6 Diagrama de flujo para la interfaz inicial del uso de la aplicación (Pantalla 1).



En la figura 3.6 se puede observar el algoritmo de la interfaz inicial de la aplicación móvil que primero ingresa a la pantalla de presentación la cual indicará una notificación de ingreso al programa de Android Studio, la cual, desplegará el botón de Ir al Mapa y dependiendo de la acción del usuario, se puede desplegar la pantalla de visualización del mapa o quedarse en la pantalla de presentación esto permite que el usuario pueda navegar entre pantallas en las cuales va a visualizar la información de la ubicación de la persona en tiempo real de acuerdo a los valores de latitud y longitud provistos por el equipo físico.

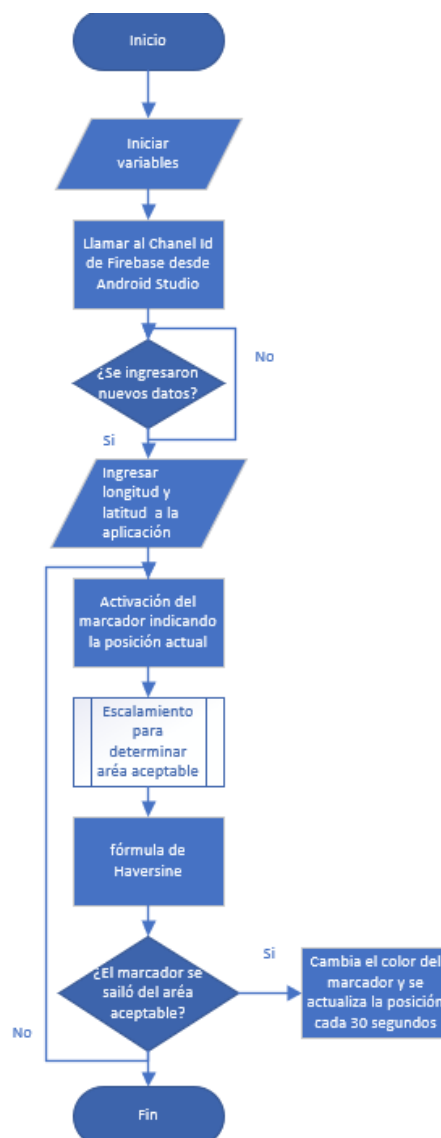
Figura 3.7 Diagrama de flujo para la interfaz inicial de la aplicación (Pantalla 1).



Elaborado por: Juan Diaz, Jimmy Rojas

En la figura 3.7 se encuentra descrito el algoritmo en el cual se desenvolverá la aplicación móvil, que comienza por el envío de un SMS al dispositivo celular y al mismo tiempo ingresa y sincroniza los datos entre Firebase y Android Studio, si se han detectado nuevas posiciones Firebase crea un nuevo canal de identificación donde se almacenaran los datos de ubicación: latitud y longitud, además de crear una notificación de puntos actualizados, caso contrario la base de datos no hará ninguna modificación, en el inicio de sus actividades, y finaliza si el usuario despliega la pantalla 2, o no genera ninguna acción reconocible por el programa.

Figura 3.8 Diagrama de flujo para la interfaz inicial del uso de la aplicación (Pantalla 2).



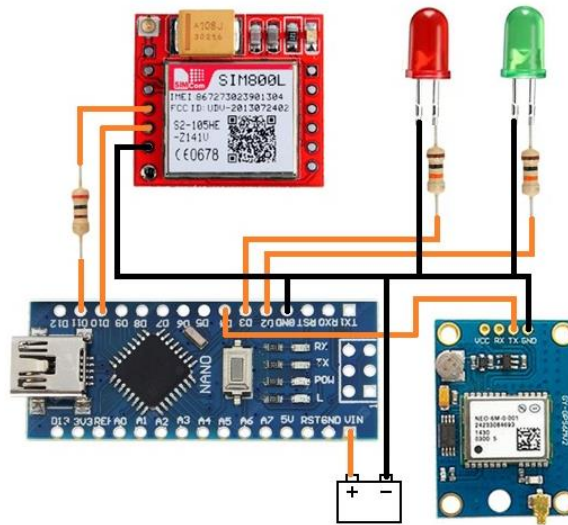
Elaborado por: Juan Diaz, Jimmy Rojas

En la figura 3.8 se muestra el funcionamiento de la aplicación en cuestión, donde el programa comienza con la iniciación de variables globales y hace un método de llamada de los datos recibidos mediante Firebase, dependiendo si reconoce una nueva ubicación, estos datos de latitud y longitud son ingresados al programa emitiendo un marcador en el servicio de Google Maps, de otro modo no hará ninguna modificación hasta que cumpla las condiciones requeridas, una vez que haya surgido el marcador ingresa a un subproceso en el cual se aplica la fórmula de Haversine para determinar si se encuentra dentro del área aceptable del programa, comparando la distancia global de las posiciones, si se cumple esta condición el marcador cambiará de color y emitirá una alerta de puntos actualizados, caso contrario entra en un bucle en el cual siempre este comparando el marcador en su posición base, en la cual se monitoreará a la persona conforme vaya cambiando su posición.

Cabe recalcar que la actualización de datos es permanente ya que es un sistema preventivo de desapariciones. Además, en el área de detección la distancia tolerable es de 30 m para alertas preventivas oportunas, esto se determinó ya que la mayoría de los hogares con índices de habitabilidad óptimos ocupan un espacio en promedio para dos personas de aproximadamente 40m^2 (Carrillo Pineda, s.f.), estas condiciones se aplican a la mayoría de las personas, en caso de que la persona de la tercera edad se encuentre en un centro de acogida o ancianato el área de aceptabilidad será de 100 a 200 m ya que estos centros y organizaciones poseen terrenos con extensiones en promedio de 2000m^2 (Calle Vidal, Guillen Coello, & Ortega Vásquez, 2010).

3.3 Diagrama Esquemático

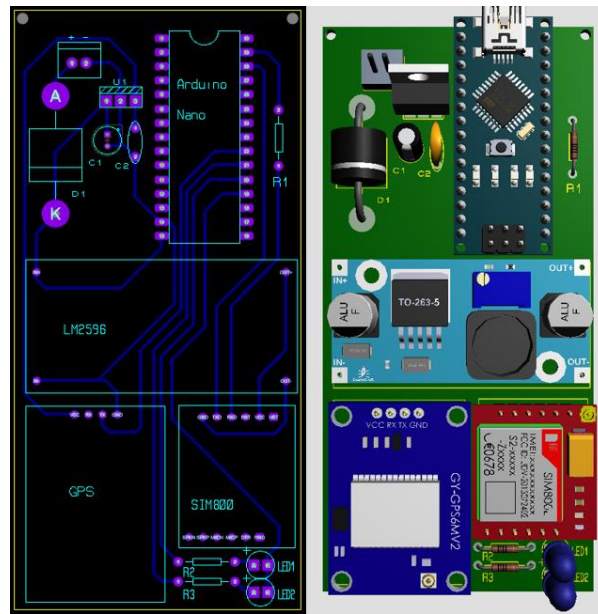
Figura 3.9 Conexiones de los módulos con la tarjeta de Arduino nano.



Elaborado por: Juan Diaz, Jimmy Rojas

La figura 3.9 muestra las conexiones de comunicación del módulo GPS y el SIM800L con el Arduino nano y un par de led que trabajan como indicadores, cada led se enciende de acuerdo a un caso específico y parpadean cuando suben el dato a la nube; Se debe tomar en cuenta los pines digitales del controlador y las configuraciones a tierra común de todos los bloques.

Figura 3.10 Diagrama Esquemático, diseñado en Proteus.

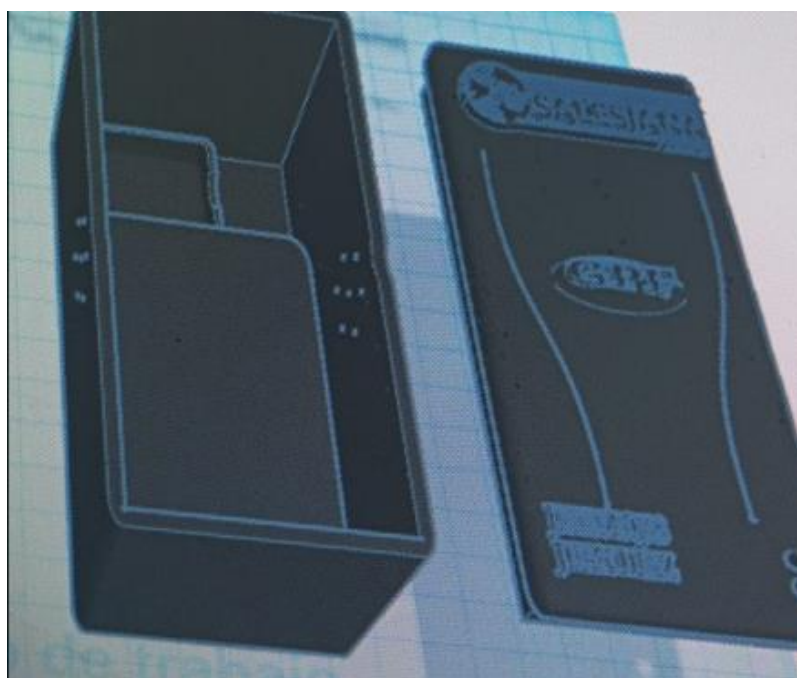


Elaborado por: Juan Diaz, Jimmy Rojas

Para las conexiones entre los módulos se construyó una placa PCB como se muestra en la figura 3.10, se diseñaron los paquetes del convertor DC-DC, GPS y del SIM800 en PCB layout debido a que los componentes no se encuentran el paquete original de Proteus, a la derecha de la figura 14 tenemos una representación 3D de la placa impresa con todos sus componentes.

Para terminar con el dispositivo se diseñó una estructura que contenga el circuito, las baterías y las antenas de los módulos GSM y GPS, estas antenas se ubicaron lo más alejados posible uno del otro para evitar cualquier tipo de interferencia y obtener mejor recepción de la señal en ambos.

Figura 3.11 Estructura del Equipo.



Elaborado por: Juan Díaz, Jimmy Rojas

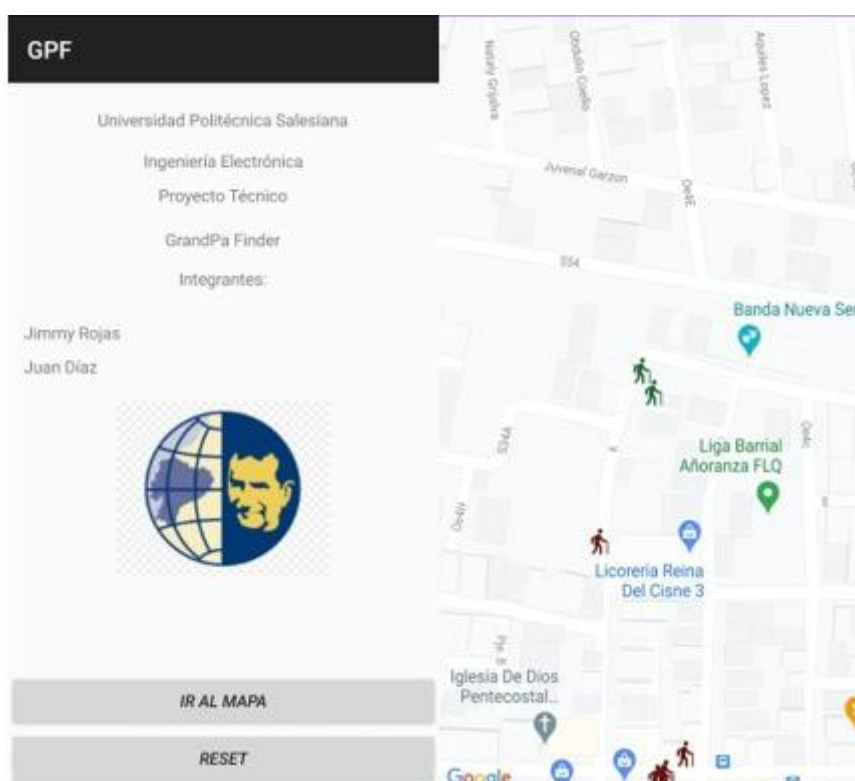
La figura 3.11 muestra la apariencia del equipo terminado con algunos detalles estéticos y ranuras de ventilación para evitar excesos de calor dentro del dispositivo y evitar el deterioro de los componentes y mal funcionamiento del mismo.

3.4 Aplicación para la prevención de desapariciones de adultos mayores.

La aplicación está desarrollada para que el usuario manipule la información de manera intuitiva, está repartida en dos interfaces o pantallas como se observa en la figura 3.12, las cuales contienen por un lado el ingreso a la aplicación que permite el inicio de la sincronización entre Firebase y Android Studio y en la segunda interfaz en donde se localiza la parte de geolocalización donde se puede observar las ubicaciones marcadas por la base de datos, esta información será de mucha importancia ya que permite que alertas tempranas preventivas y monitoreo en vivo del movimiento de la persona de la tercera edad haciendo que su riesgo sea el menor posible. Esto es posible ya que los módulos que están enlazados con Arduino y estos envían su información mediante peticiones HTTP y comandos AT, los cuales en Firebase se transforman en archivos

JSON que son codificados y enviados a Android Studio, Android Studio recopila todos estos archivos codificados y filtra por código los datos de interés y con los Servicios de Google Maps, Geolocalización, Geocodificación e identificadores de lugares, permiten en la interfaz marcar las posiciones.

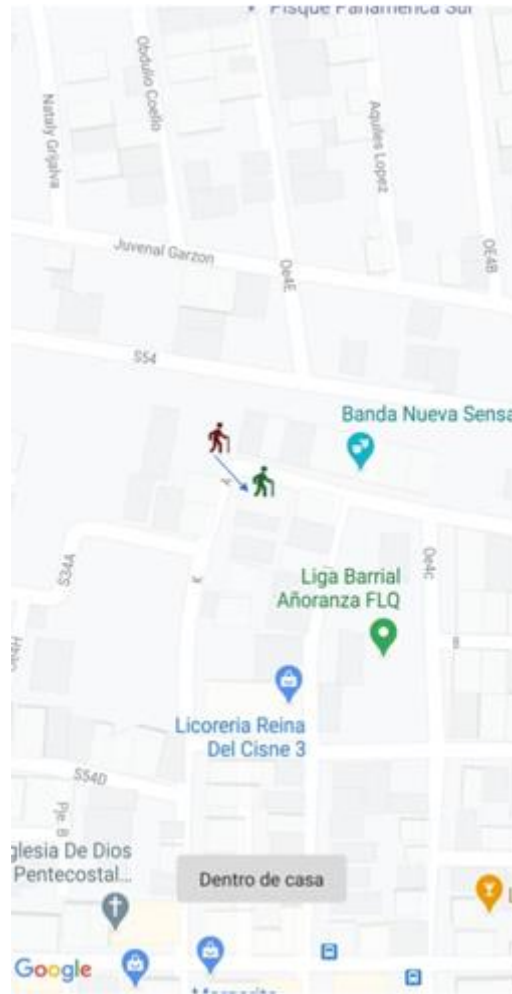
Figura 3.12 Interfaces de la aplicación desarrollada en Android Studio.



Elaborado por: Juan Díaz, Jimmy Rojas

Esta aplicación permite que la latitud y longitud tomadas, ingresen a un filtro por código en el cual se aíslan los valores en un arreglo de vectores y mediante la fórmula de Haversine medir la distancia que se conformara parte del radio del área de aceptabilidad, que tomara estos datos y dependiendo si la persona en cuestión abandona o ingresa al área de aceptabilidad el marcador que indica su posición cambiara de color emitiendo alertas de actualización de puntos como se observa en la figura 3.12.

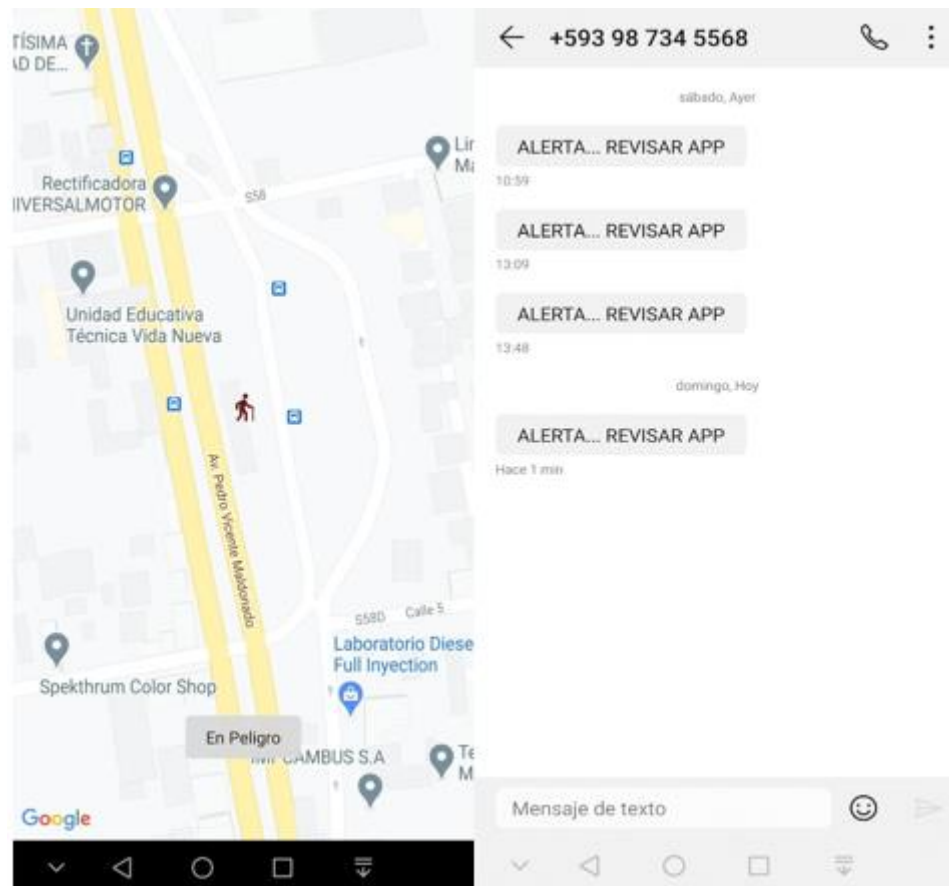
Figura 3.13 Estado de la persona: Dentro de casa.



Elaborado por: Juan Diaz, Jimmy Rojas

Una de las alertas emergentes dentro de la interfaz de la aplicación consta de un mensaje que se despliega cuando la persona reingresa o se encuentra en la misma posición reiteradamente dentro del área de aceptabilidad como se observa en la figura 3.13, este mensaje se auto cancela dentro de 3 segundos hasta detectar un cambio de posición del equipo.

Figura 3.14 Estado de la persona: En peligro.



Elaborado por: Juan Diaz, Jimmy Rojas

Como se aprecia en la figura 3.14 en el estado En peligro se determina cuando la persona está fuera del área de aceptabilidad y se despliega un mensaje de texto a un número de celular registrado que corresponde a la persona a cargo del cuidado del adulto mayor, además en la interfaz de la aplicación se despliega un mensaje con las mismas características del apartado anterior. Cabe tomar en consideración que el mensaje de texto se vuelve a enviar cada vez que se hayan registrado 30 entradas de datos en nuestra base de datos en tiempo real en la plataforma digital Firebase.

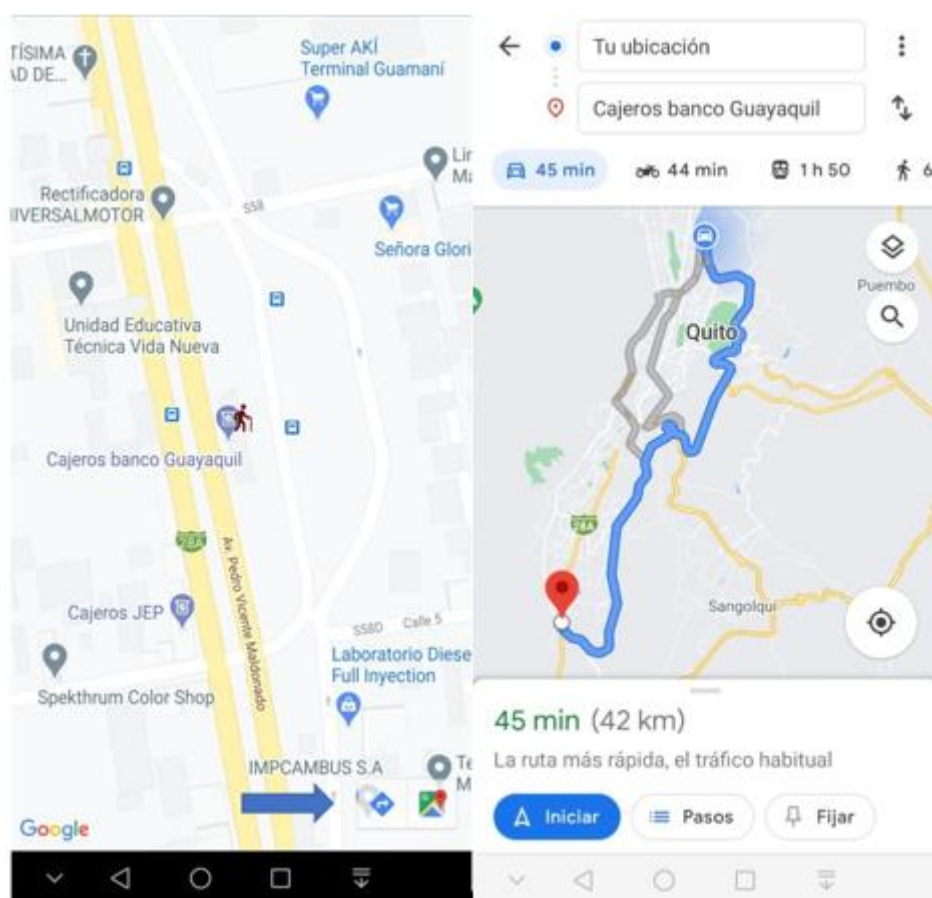
Figura 3.15 Estado de la persona: Actualizando.



Elaborado por: Juan Diaz, Jimmy Rojas

Si la persona se encuentra en un lugar en el cual la señal del equipo no sea la adecuada o existe algún problema de conexión, se despliega un mensaje con las mismas características de los apartados anteriores, siempre recalcando el último punto enviado por el equipo como muestra la figura 3.15.

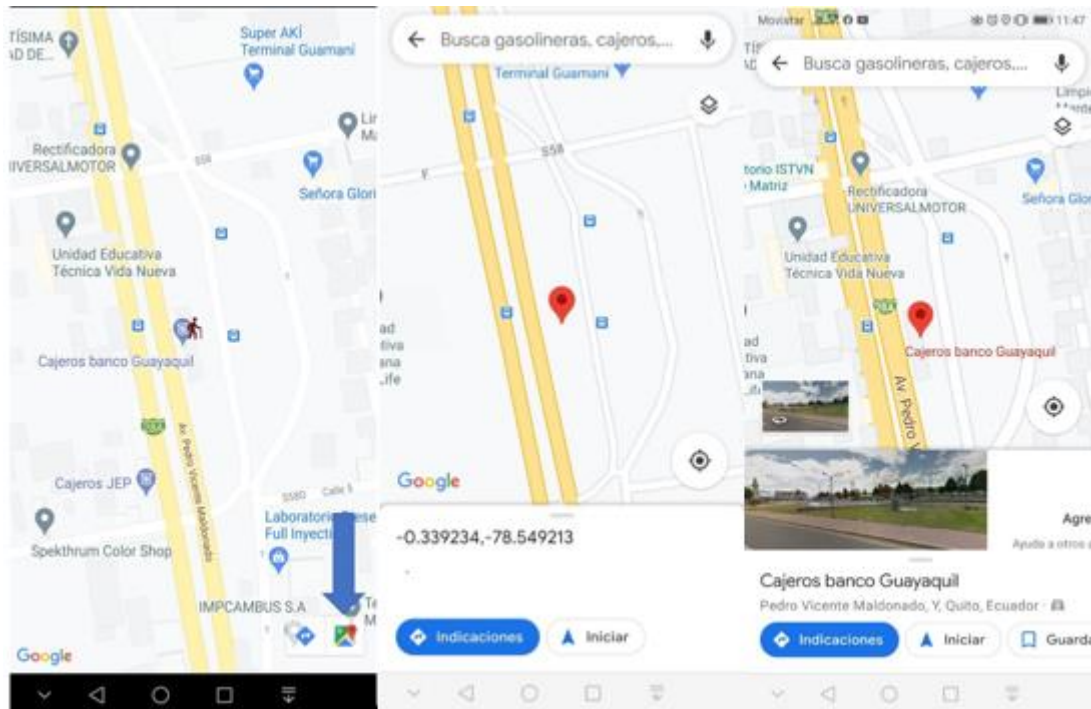
Figura 3.16 Vinculación con Servicios de rutas Google Maps.



Elaborado por: Juan Diaz, Jimmy Rojas

Ya que Android Studio cuenta con el apoyo de Google en el desarrollo de aplicaciones la integración de los servicios de Google Play aplica para nuestro equipo, permitiendo que los servicios de rutas de Google Maps se desplieguen una vez se selecciona el marcador de interés donde se encuentre la persona, y así poder determinar una trayectoria en la cual podamos llegar al destino deseado en el menor tiempo posible desde nuestra ubicación en el mapa como se observa en la figura 3.16.

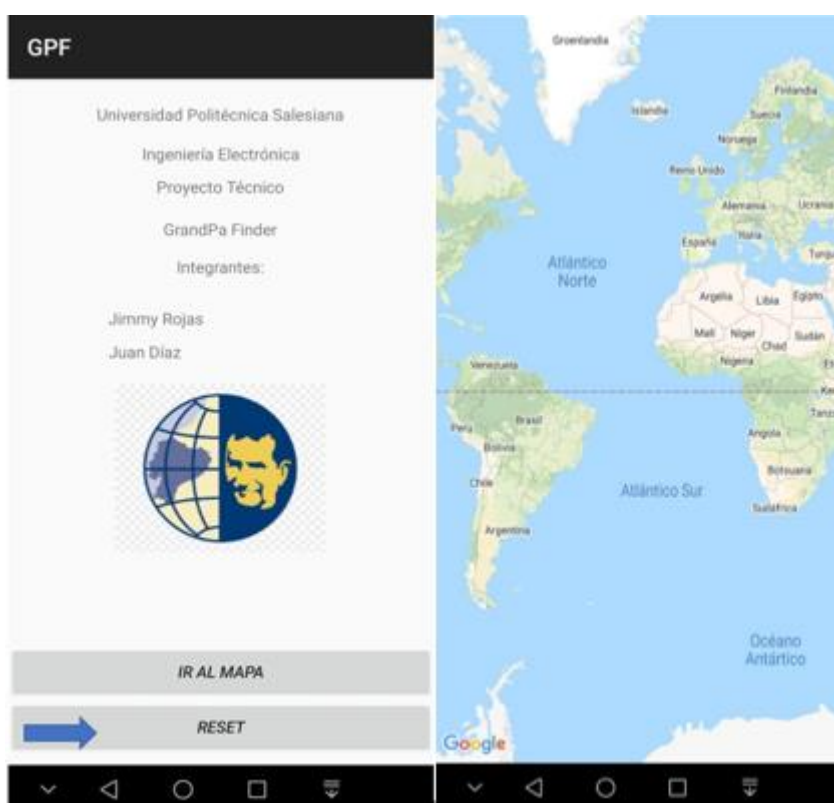
Figura 3.17 Vinculación con Servicios de lugares de Google Maps.



Elaborado por: Juan Diaz, Jimmy Rojas

Otro servicio integrado a la aplicación de nuestro equipo nos permite observar un lugar de referencia gracias a los servicios de lugares de Google Maps así como sus coordenadas de latitud y longitud, permitiendo un acercamiento más preciso a un lugar de interés de la persona en cuestión, esta herramienta nos brinda lugares de referencia como plazas, locales comerciales, carreteras, estaciones de servicios, referencias geográficas, incluyendo una visión en 3D del último lugar captado por el dispositivo como se observa en la figura 3.17.

Figura 3.18 Finalización del ciclo de monitoreo.



Elaborado por: Juan Diaz, Jimmy Rojas

Una vez terminado el monitoreo del adulto mayor ya sea en una situación de peligro o una situación favorable, se recomienda al usuario la función reset para iniciar un nuevo ciclo de monitoreo limpiando los marcadores registrados hasta ese punto, sin necesidad de resetear al equipo físico, una vez realizado este proceso la interfaz de la aplicación se limpia mostrando el mapa completo como se muestra en la figura 3.18, una vez que se registre una posición nueva la cámara de la aplicación se dirigirá automáticamente a la nueva posición como se observa en la figura 3.13.

3.5 Geolocalización de la persona de la tercera edad

Mencionamos anteriormente que la distancia para el área de aceptabilidad se determina con la fórmula de Haversine que se utiliza para calcular la distancia entre dos puntos en el plano geográfico (Amaya Jaramillo & Carrera Chicaiza, 2010).

$$haversin\left(\frac{d}{R}\right) = haversin(\phi_s - \phi_f) + \cos(\phi_s) \cos(\phi_f) haversin(\Delta\lambda) \quad \text{Ec. (3.1)}$$

Donde:

- $haversine(\theta) = \sin^2\left(\frac{\theta}{2}\right)$
- d es la distancia entre los dos puntos.
- R es el radio de la esfera
- ϕ_s es la latitud del punto 1
- ϕ_f es la latitud del punto 2
- $\Delta\lambda$ es la diferencia de longitudes de los puntos

Tomando como parámetro “d” como la constante que se va a comparar en el programa de Arduino y Android Studio para determinar si la persona adulta mayor está en riesgo.

3.6 Resultados Obtenidos.

Para el análisis de la precisión y exactitud del dispositivo se tomó un tamaño de la muestra mediante la siguiente formula, que nos permite calcular el número de muestras a tomar sin considerar la población: (Bernal Torres, 2010)

$$n = \frac{z^2 pq}{E^2} \quad \text{Ec. (3.2)}$$

n: Tamaño de la muestra que se va a estimar.

z: Parámetro estadístico de confianza.

p: Proporción donde la nueva posición respecto a la referencia sea menor o igual a 5m.

q: Proporción donde la nueva posición respecto a la referencia sea mayor a 5m.

E: Error estimado.

Para la formula se tomó una probabilidad de confianza del 95%, es decir, valor de confianza $z = 1.96$ y un error estimado del 5%, la probabilidad de obtener una distancia menor a 5m del 98% y la probabilidad de obtener distancia mayor a 5m del 2%.

$$n = \frac{z^2 pq}{E^2}$$

$$n = \frac{1.96^2 * 0.98 * 0.02}{0.05^2}$$

$$n = 30$$

Con el nivel de confianza del 95% y un error estimado al 5% se necesitan de 30 muestras para determinar qué tan exacto y preciso es el dispositivo.

Tabla 3.1 Mediciones de exactitud y precisión del dispositivo.

Muestra	Latitud	Longitud	Hora	Distancia hacia la referencia*
1	-0.330702	-78.55835	14:31:20	5.01m
2	-0.330696	-78.55835	14:34:28	4.75 m
3	-0.330681	-78.558319	14:37:36	0.97 m
4	-0.330712	-78.558334	14:40:44	4.53 m
5	-0.330706	-78.558319	14:43:53	3.30 m
6	-0.330679	-78.558304	14:47:01	0.72 m
7	-0.330715	-78.558327	14:50:09	4.30 m
8	-0.330701	-78.558327	14:53:17	3.10 m
9	-0.330701	-78.558327	14:56:25	3.10 m
10	-0.330686	-78.558311	14:59:33	0.75 m
11	-0.330704	-78.558311	15:02:41	4.49 m
12	-0.330667	-78.558311	15:05:50	1.42 m
13	-0.330726	-78.558311	15:08:58	5.39 m
14	-0.330717	-78.558319	15:12:06	4.63 m
*15	-0.330678	-78.558311	15:15:14	0 m
16	-0.330687	-78.558319	15:18:22	1.51 m
17	-0.330651	-78.558296	15:21:30	3.10 m
18	-0.330642	-78.558319	15:24:38	3.89 m
19	-0.330695	-78.558296	15:27:47	2.64 m
20	-0.330707	-78.558327	15:30:55	3.74 m

21	-0.330685	-78.558334	15:33:40	2.79 m
22	-0.330656	-78.558357	15:36:56	5.64 m
23	-0.330664	-78.558342	15:39:58	3,71 m
24	-0.330692	-78.557316	15:42:14	2.52 m
25	-0.330682	-78.558304	15:45:32	0.93 m
26	-0.330663	-78.558292	15:48:15	2.97 m
27	-0.330673	-78.557337	15:51:25	3.01 m
28	-0.330684	-78.558311	15:54:34	0.34 m
29	-0.330665	-78.558292	15:57:36	2.84 m
30	-0.330672	-78.558326	16:00:05	1.91 m
31	-0.330689	-78.558295	16:03:09	2.51 m

Elaborado por: Los Autores.

Como se puede observar en la tabla 3.1 para la prueba de precisión y exactitud se tomó 30 muestras de la posición sobre el punto de referencia*, en el cual constan la latitud, longitud, la hora, y que distancia hubo hacia la referencia en metros. A continuación, utilizamos la ecuación del error absoluto para determinar el rango confiable en metros del dispositivo.

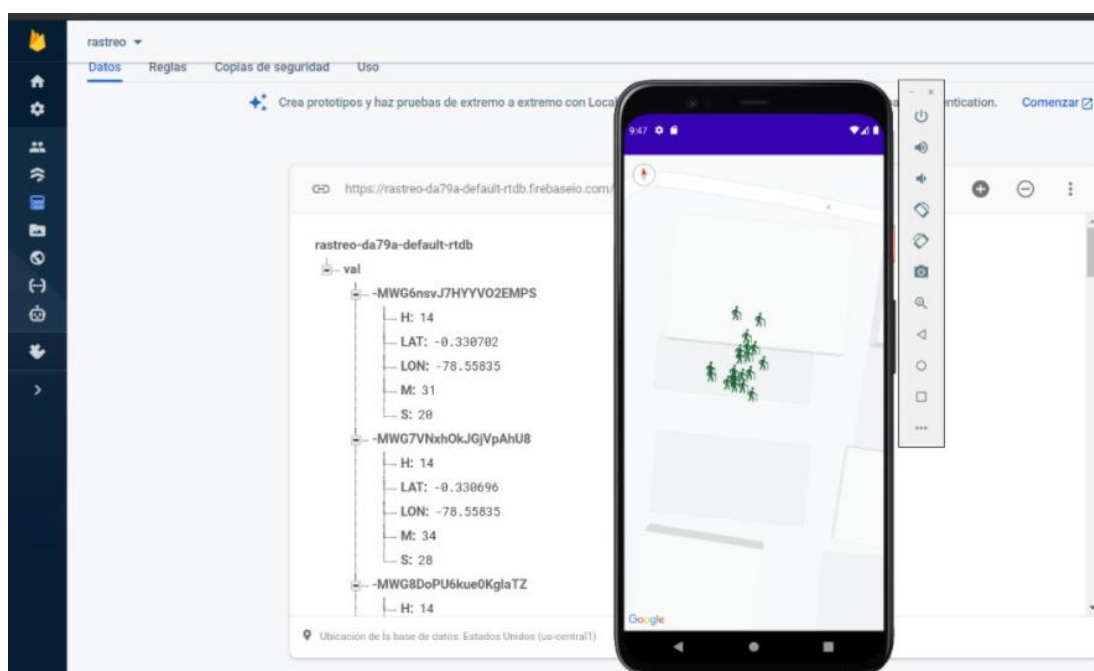
$$\varepsilon = \sqrt{\frac{\sum(x-\bar{x})^2}{N(N-1)}} \quad \text{Ec (3.3)}$$

El promedio de las muestras de distancia respecto al punto de referencia es de $\bar{x} = 3.017m$ la cual representa nuestro valor exacto de las muestras tomadas y procedemos a calcular el error absoluto con la ecuación 2.

$$\varepsilon = \sqrt{\frac{\sum(x - \bar{x})^2}{N(N - 1)}} = \sqrt{\frac{62.965}{30(30 - 1)}} = 0.269$$

El error de medida del dispositivo respecto al punto de referencia tomado gira en torno a los 3.017 ± 0.269 m.

Figura 3.19 Análisis de precisión y exactitud de la posición del dispositivo.



Elaborado por: Juan Diaz, Jimmy Rojas

En la figura 3.19 se puede apreciar la manera en que se distribuyen los parámetros recopilados en la tabla 3.1, y se puede constatar que la precisión de los datos está en un rango admisible para las condiciones del dispositivo, para el dato de referencia se tomó en consideración un punto estratégico del lugar en el que se realizó la prueba en este caso en el área social de la casa que se puede reconocer como la parte gris de un color más oscuro, que cuenta con un espacio libre en el cual pueda el módulo GSM conectarse a la red y enviar los datos.

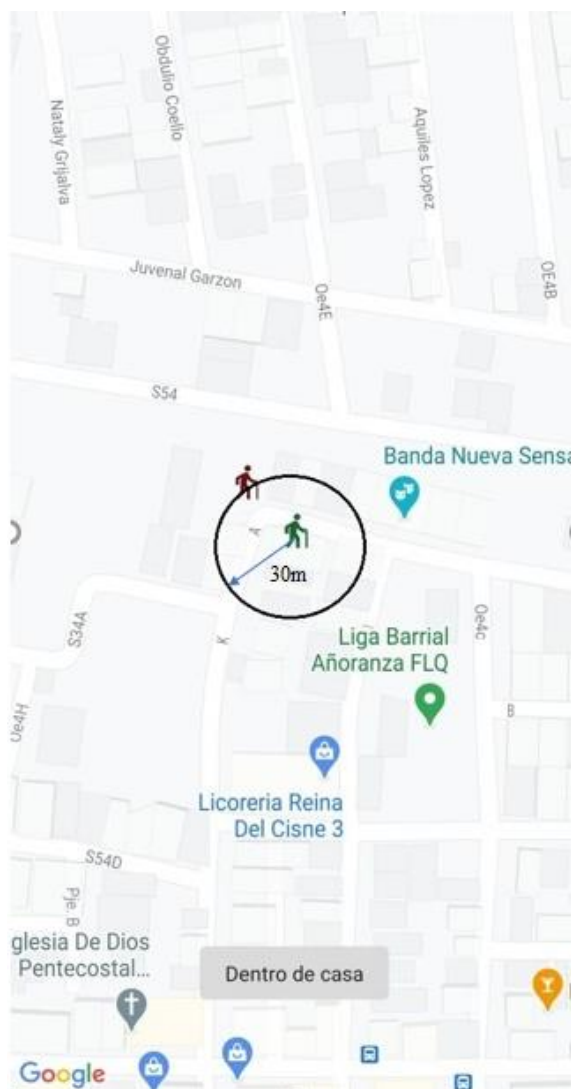
Figura 3.20 Prueba de Campo.



Elaborado por: Juan Diaz, Jimmy Rojas

Una vez determinada la precisión y exactitud del dispositivo se realizó una prueba en tiempo real, en un trayecto que se puede visualizar en la Figura 3.20, esta prueba se demoró 1 hora, 49 minutos y 19 segundos, en los cuales el dispositivo respondió de manera correcta y sin inconvenientes en la transmisión de datos, en el cual se pudo comprobar el área de aceptabilidad y se determinó de una manera certera la posición del sujeto de prueba en todo momento.

Figura 3.21 Área aceptable.



Elaborado por: Juan Diaz, Jimmy Rojas

Dentro los últimos parámetros del desarrollo del proyecto se precisó el área de aceptabilidad como se observa en la figura 3.21, tomando en cuenta la casa de prueba y la distancia tolerable de acuerdo a parámetros revisados en apartados anteriores, con los cuales se calibró las alertas preventivas y alertas auxiliares del sujeto en peligro.

CAPÍTULO 4

ANÁLISIS DE COSTOS

En este capítulo se detalla el análisis de costos de los módulos y elementos que son parte del desarrollo del dispositivo, este estudio determinará si el dispositivo es factible para uso en adultos mayores. Las variables de evaluación del proyecto para su viabilidad serán el costo fijo, costo variable y el punto de equilibrio.

4.1 Costos de Materiales y Módulos

Para la construcción del dispositivo se inició con la búsqueda de módulos que cumplan las necesidades propuestas, bajo costo y que sean compatibles con el microcontrolador.

Tabla 4.1 Materiales y Módulos.

Equipo	Cantidad	Unitario	Total
Módulo GSM SIM800	1	\$10,99	\$10,99
Chip móvil	1	\$3	\$3
Modulo GPS NEO 6m	1	\$10,50	\$10,50
Conversor Buck	1	\$4,02	\$4,02
Arduino Nano	1	\$9	\$9
Baterías 186500	2	\$4	\$8
Resistencia 10K ohms	2	\$0,10	\$0,20
Resistencia 20K ohms	1	\$0,10	\$0,10
Diodo 6A10	1	\$0,27	\$0,27
Regulado LM7805	1	\$0,45	\$0,45
Capacitores	2	\$0,13	\$0,26
LED's	2	\$0,18	\$0,36
Placa PCB	1	\$22	\$22
Estructura	1	\$30	\$30
		<u>TOTAL</u>	\$99,15

Elaborado por: Los Autores.

En la Tabla 4.1 se detallan los recursos necesarios utilizados para el desarrollo del dispositivo, se consideró módulos que sean de fácil adquisición y de bajo costo.

4.2. Indicadores para la factibilidad del Proyecto

Al inicio del capítulo 4 se mencionó los indicadores que se toman en cuenta para decidir la posibilidad de instauración del proyecto.

Los valores finales que se toman en cuenta para el proyecto se observan en la tabla 5.

Tabla 4.2 Precio Total del Proyecto.

Inversión	Valor
Equipo	\$ 99.15
Servicio Móvil	\$7
Servicio Nube	\$5
TOTAL	\$ 111.15

Elaborado por: Los Autores.

La Tabla 4.2 refleja el costo final del proyecto conjunto con los servicios de telefonía móvil para la conexión a la red y a la base de datos Firebase.

4.2.1. Costo Fijo

“Son aquellos costos que permanecen constantes durante un periodo de tiempo determinado, sin importar el volumen de producción”. (Universidad Veracruzana, 2013)

4.2.2. Costo Variable.

“Son aquellos que se modifican de acuerdo con el volumen de producción, es decir, si no hay producción no hay costos variables y si se producen muchas unidades el costo variable es alto. Unitariamente el costo variable se considera Fijo, mientras que en forma total se considera variable.” (Universidad Veracruzana, 2013)

4.2.3. Punto de Equilibrio

“El punto de equilibrio es aquel nivel de actividad en el que la empresa consigue cubrir la totalidad de sus costes, tanto fijos como variables, obteniendo un beneficio cero.” (Universidad Veracruzana, 2013)

$$Q_E = \frac{C_F}{(P - C_v)} \quad \text{Ec (4.1)}$$

$Q_E \rightarrow$ Punto de Equilibrio.

$C_F \rightarrow$ Costo fijo.

$C_v \rightarrow$ Costo variable.

$P \rightarrow$ Precio unitario.

El punto de equilibrio para el proyecto se determinó con los valores de la tabla 5 de la siguiente manera.

$$Q_E = \frac{99,15}{(120 - 12)} = 0.918$$

De acuerdo al cálculo anterior el dispositivo es viable en precio y competitivo en el mercado ya que en promedio un dispositivo de características similares ronda un precio promedio de \$164.33 llevando un margen de 8.2% de ganancia por equipo.

CAPÍTULO 5

CONCLUSIONES

Se implementó un dispositivo de monitoreo IOT y GPS para determinar la posición de la persona adulta mayor calculando una superficie determinada como área aceptable o segura, donde, si llega a salirse del área calculada se envía mensajes de alerta usando la red móvil e informando que la persona a su cuidado se está alejando del sitio seguro, para que tomen las medidas respectivas.

Se desarrollo una base de datos en tiempo real en la plataforma digital de Firebase que permitió el almacenamiento de las posiciones determinadas por el equipo, en el cual se envió la hora exacta en la cual la persona en riesgo está en un punto determinado permitiendo valorar el tiempo que permaneció en un lugar en un ciclo de monitoreo.

Se verifico el correcto funcionamiento de acuerdo a los parámetros de habitabilidad mencionados en apartados anteriores estableciendo un área circular de 30m de radio para el área de aceptabilidad, se obtuvo un error en la precisión de $3.017 \pm 0.269\text{m}$ de un total de 30 muestras de la misma ubicación por lo que se concluye que la precisión del dispositivo es aceptable.

Dentro del capítulo de análisis de costo se aplicaron indicadores de factibilidad tanto costo fijo como costo variable, con los cuales se determinó un Punto de equilibrio en 0.918, que nos da un margen de ganancia de 8.2% por dispositivo, concluyendo que el equipo es factible para la implementación para adultos mayores.

RECOMENDACIONES

Para futuras mejoras sobre el trabajo realizado se puede implementar un módulo de mayor costo y con tecnología 5G que agilizará el envío de los datos y reducirá el tiempo en establecer conexión con el servidor.

Es importante recalcar que el equipo desarrollado es de dimensiones acorde al presupuesto determinado en el análisis de costos, estas dimensiones se pueden reducir con técnicas de microelectrónica que se recomienda para la comercialización de este equipo.

Sobre el uso de la aplicación se recomienda la implementación de algún algoritmo que permita optimizar la autenticación de los usuarios del equipo, así como, definir bajo sus propios parámetros la utilización personalizada de cada dispositivo en caso de fabricación en volumen, añadiendo sincronizaciones con las cuentas del usuario para que las alertas se generen por medio de correo-electrónico.

REFERENCIAS BIBLIOGRÁFICAS

- Amaya Jaramillo, C. D., & Carrera Chicaiza, D. P. (Julio de 2010). *Sistema de Monitoreo Pasivo para Vehículos mediante GPS*. Recuperado el 2021, de Escuela Politécnica Nacional: <https://bibdigital.epn.edu.ec/bitstream/15000/2376/1/CD-3108.pdf>
- Bernal Torres, C. A. (2010). *Metodología de la investigación*. Bogotá D.C., Colombia: PEARSON.
- Calle Vidal, C. C., Guillen Coello, L. A., & Ortega Vásquez, X. E. (2010). *Proyecto de Inversión Empresarial Guardería de Ancianos "Jardines Años Dorados"*. Recuperado el 2021, de Universidad Politécnica Salesiana: <https://dspace.ups.edu.ec/bitstream/123456789/1368/14/UPS-CT002233.pdf>
- Cardona, M. P. (14 de octubre de 2019). *Escuela de Negocios de la Innovación y los Emprendedores(IEBS)*. Obtenido de Firebase: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Carrillo Pineda, A. P. (s.f.). *Diseño Alternativo de Vivienda Óptima*. Recuperado el 2021, de Universidad Internacional SEK: <https://repositorio.uisek.edu.ec/bitstream/123456789/225/6/ESCRITO.pdf>
- Chango Quinapanta, S. (2013). *Repositorio Digital de la Universidad Técnica de Ambato*. Obtenido de Utilización del Cloud Computing para mejorar el aprendizaje colaborativo en los/las Estudiantes de la Carrera De Docencia En Informática De La Facultad De Ciencias Humanas y De La Educación De La Universidad Técnica De Ambato: <https://repositorio.uta.edu.ec/jspui/bitstream/123456789/6184/1/FCHE-INFOR-40.pdf>
- Developers Android. (s.f.). *Developers*. Obtenido de Introducción a Android Studio: <https://developer.android.com/studio/intro?hl=es-419>
- Electronics, N. (2016). *Naylamp*. Obtenido de Modulo GPS Arduino: https://www.naylampmechatronics.com/blog/18_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html
- Electronics, N. (2020). *Naylamp*. Obtenido de Módulo GSM SIM800L: <https://www.naylampmechatronics.com/inalambrico/115-modulo-gsm->

sim800l-2g.html

LEXIS. (2013). *Protocolo Actuación Búsqueda Investigación de Personas Desaparecidas*. Obtenido de LEXIS: <https://www.policia.gob.ec/wp-content/uploads/downloads/2019/10/PROTOCOLO-ACTUACION-BUSQUEDA-INVESTIGACION-DE-PERSONAS-DESAPARECIDAS.pdf>

López, S. (17 de Mayo de 2020). *Digital 55*. Obtenido de Firebase: qué es, para qué sirve, funcionalidades y ventajas: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/#:~:text=%C2%BFPara%20qu%C3%A9%20sirve%20Firebase%3F,renunciar%20a%20la%20calidad%20requerida>

Luna, V. (Septiembre de 2020). *Universidad Andina Simon Bolívar*. Obtenido de El Internet de las cosas (IoT) y la protección de la sombra digital: <https://www.uasb.edu.ec/web/ciberderechos/el-internet-de-las-cosas-y-la-proteccion-de-la-sombra-digital>

Master. (2020). *Shop Master*. Obtenido de Arduino Nano: <https://shop.master.com.mx/product/detail?id=7454>

Pérez, E. (26 de Octubre de 2017). *Ataka Android*. Obtenido de Android Studio 3.0: ya disponible la versión estable con soporte para Kotlin y las nuevas APIs de Android: <https://www.xatakandroid.com/programacion-android/android-studio-3-0-ya-disponible-la-version-estable-con-soporte-para-kotlin-y-las-nuevas-apis-de-android>

robótica, e.-i. e. (s.f.). *e-ika electrónica y robótica*. Obtenido de Módulo GPS NEO GY-GPS6MV2 para Arduino : <https://www.e-ika.com/modulo-gps-gy-gps6mv2-para-arduino-2>

Universidad Veracruzana. (Febrero de 2013). *Clasificación de los Costos*. Obtenido de <https://www.uv.mx/personal/alsalas/files/2013/02/CLASIFICACION-DE-LOS-COSTOS.pdf>

Wevar, J. A. (2005). *Universidad Austral de Chiles*. Obtenido de Análisis y Estudio de Redes GPRS: <http://cybertesis.uach.cl/tesis/uach/2005/bmfcis211a/doc/bmfcis211a.pdf>

ANEXOS

ANEXO 1. Código de Programación en Arduino.

Anexo 1.1 Código de inicialización de librerías y variables.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

#define LEDv 2
#define LEDr 3

boolean valido=false;

String datos=" ";

unsigned long t_gps_i=0;
unsigned long t_gps=0;
unsigned long tiempo1=0;
unsigned long tiempo2=0;
unsigned long tiempo3=0;
unsigned long tiempo4=0;

int caso=0,contMuestra=0, muestreo=0, sms=0, t=50;

float La_i=-0.330523, Lo_i=-78.558309;
float La_f, Lo_f, h, m, s, restal, resta2, aux, d;
```

Anexo 1.2 Código donde se establece los pines y se inicializa la Comunicación Serial, además la configuración básica del módulo GSM.

```
SoftwareSerial gpsSerial(4, 5); //pines de comunicación GPS
SoftwareSerial gsmSerial(10, 11); //pines de comunicación GSM

void setup()
{
  pinMode(LEDv,OUTPUT);
  pinMode(LEDv,OUTPUT);
  Serial.begin(9600);
  gsmSerial.begin(9600);
  Serial.println("configuración inicial");
  delay(10000);
  gsmSerial.listen();
  config_gsm_inicial();
  gpsSerial.begin(9600);
}
```

Anexo 1.3 Código donde se validan los datos del GPS.

```
void loop()
{
    // Comprueba que sean datos validos entregados por el GPS.
    while (gpsSerial.available() > 0)
        if (gps.encode(gpsSerial.read()))

            if(gps.location.lat()!=0 && gps.location.lng()!=0)
                valido=true;
            else
                valido=false;

    if(valido==true)
        inicio();
}
```

Anexo 1.4 Código donde se calcula el parámetro de escalamiento para determinar el área de aceptabilidad.

```
void inicio()
{
    digitalWrite(LEDv,HIGH);
    gpsInfo();
    resta1=La_f - La_i;
    resta2=Lo_f - Lo_i;
    aux=sqrt(pow(resta1,2)+pow(resta2,2));
    d=aux*10000; //variable de escalamiento que determina el área aceptable
}
```

Anexo 1.5 Código donde realiza 30 lecturas.

```
if(contMuestra>=5)
{
    t_gps=millis();
    while(millis()-t_gps<=100);
    contMuestra=5;
    muestreo++;
}
```

Anexo 1.6 Código donde toma la decisión de realizar una actividad u otra.

```
if(d<=3)
    caso=0;
else
    caso=1;
```

Anexo 1.7 Código donde verifica que se realizó 30 lecturas y ejecuta acciones dependiendo de si está dentro o fuera del área de aceptabilidad.

```

if(muestreo>=30)
    switch(caso)
    {
        case 0:
            digitalWrite(LEDr,LOW);
            envioDato();
            gpsSerial.listen();
            while(gpsSerial.available() > 0) datos = Serial.read(); //limpio el buffer
            digitalWrite(LEDv,HIGH);
            tiempo3=millis();
            while(millis()-tiempo3<=120000);
            break;

        case 1:
            digitalWrite(LEDr,HIGH);
            envioDato();
            if (sms==0 || sms==30)
                envioMensaje();
            sms++;
            gpsSerial.listen();
            while(gpsSerial.available() > 0) datos = Serial.read(); //limpio el buffer
            digitalWrite(LEDv,HIGH);
            digitalWrite(LEDr,HIGH);
            tiempo4=millis();
            while(millis()-tiempo4<=20000);
            break;
    }
}

```

Anexo 1.8 Código donde se configura parámetros del módulo GSM y se configura la dirección URL de nuestra base de datos de Firebase.

```

void config_gsm_inicial()
{
    while(enviarAT("AT", "OK", 2000)==0);
    while(enviarAT("AT+CREG?", "+CREG: 0,1", "+CREG: 0,5", 2000) == 0);
    while(enviarAT("AT+SAPBR=3,1,\"Contype\",\"GPRS\"", "OK", 2000)==0);
    while(enviarAT("AT+SAPBR=3,1,\"APN\",\"internet.movistar.com.ec\"", "OK", 2000)==0);
    while(enviarAT("AT+SAPBR=1,1,\"OK\", 35000) == 0);
    while(enviarAT("AT+SAPBR=2,1\", \"OK\", 2000)==0); //obtiene la ip
    while(enviarAT("AT+HTTIPINIT\", \"OK\", 2000)==0);
    while(enviarAT("AT+HTTIPARA=\"CID\",1\", \"OK\", 2000)==0);
    while(enviarAT("AT+HTTIPARA=\"URL\", \"", \"OK\", 4000)==0);
}

```

Anexo 1.9 Código encargado de subir los datos y de ejecutar el parpadeo de leds para informar que se subió el dato a la nube.

```
void envioDato()
{
    gsmSerial.println("AT+HTTPDATA=100,7000");
    datos="{\"LAT\": \""+String(La_f, 6)+"\", \"LON\": \""+String(Lo_f, 6)+"\", \"H\": \""+String(h, 0)
    gsmSerial.println(datos);
    tiempo1=millis();
    while(millis()-tiempo1<=8000);
    gsmSerial.println("AT+HTTPACTION=1");
    datosSerial();
    tiempo2=millis();
    while(millis()-tiempo2<=55000);
    datosSerial();
    indicador();
    muestreo=0;
}
```

Anexo 1.10 Código para el envío del SMS a la persona encargada alertando que la persona en cuestión acaba de abandonar o salir del área de aceptabilidad y que revise la aplicación para que tomen las medidas respectivas de su localización.

```
void envioMensaje()
{
    while(enviarAT("AT+CMGF=1", "OK", 2000)==0);
    gsmSerial.println("AT+CMGS=\"+593[REDACTED]\""); //número al cual se enviará el sms
    delay(500);
    gsmSerial.print("ALERTA... REVISAR APP");
    delay(500);
    gsmSerial.write(26);
    datosSerial();
}
```

Anexo 1.11 Código que valida la respuesta de los comandos AT antes de ejecutar el siguiente comando AT.

```
//Metodo que valida si la respuesta del comando AT es correcto//
int enviarAT(String comandoAT, const char* resp_correcta, unsigned int espera)
{
    int x = 0;
    bool correcto = 0;
    char respuesta[100];
    unsigned long anterior;
    memset(respuesta, '\0', 100); //Inicializa el string
    delay(100);
    while ( gsmSerial.available() > 0) gsmSerial.read(); //Limpia el buffer de entrada
    gsmSerial.println(comandoAT); //Envia el comando AT
    x = 0;
    anterior = millis();
    // Espera una respuesta
    do {
        // si hay datos el buffer de entrada del UART lee y comprueba la respuesta
        if (gsmSerial.available() != 0)
        {
            respuesta[x] = gsmSerial.read();
            x++;
            // Comprueba si la respuesta es correcta
            if (strstr(respuesta, resp_correcta) != NULL)
            {
                correcto = 1;
            }
        }
    }

    }while ((correcto == 0) && ((millis() - anterior) < espera)); // Espera hasta tener una respuesta
    return correcto; // Retorna verdadero o falso
}
```

Anexo 1.12 Código donde obtiene los datos de posición y hora.

```
void gpsInfo()
{
    if (gps.location.isValid())
    {
        La_f=(gps.location.lat());
        Lo_f=(gps.location.lng());
    }
    else
        Serial.print(F("Datos invalidos"));

    if (gps.time.isValid())
    {
        h=(gps.time.hour()-5);
        m=gps.time.minute();
        s=gps.time.second();
    }
    else
        Serial.print(F("Datos invalidos"));
}
```

Anexo 1.13 Código para la ejecución del parpadeo de leds para informar que el dato ha sido subido.

```
void indicador()
{
    digitalWrite(LEDv, LOW);
    digitalWrite(LEDd, LOW);
    for(int j=0; j<=3; j++)
    {
        digitalWrite(LEDv, HIGH);
        delay(t);
        digitalWrite(LEDd, LOW);
        delay(t);
        digitalWrite(LEDv, LOW);
        delay(t);
        digitalWrite(LEDd, HIGH);
        delay(t);
    }
    digitalWrite(LEDv, LOW);
    digitalWrite(LEDd, LOW);
}
```

Anexo 1.14 Método para usar el puerto serial por software.

```
void datosSerial()
{
    while (Serial.available())
        gsmSerial.write(Serial.read());

    while(gsmSerial.available())
        Serial.write(gsmSerial.read());
}
```

ANEXO 2. Código de Programación del GPF en Android Studio.

Anexo 2.1 Código para la interfaz principal (Pantalla 1).

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private Button mBtnMaps;
    private Button mBtnGpf;
    private DatabaseReference mDatabase;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mBtnMaps= findViewById(R.id.gpfd);
        mBtnMaps.setOnClickListener(this);
        mBtnGpf=(Button) findViewById(R.id.gpf);
        mDatabase= FirebaseDatabase.getInstance().getReference();
        mBtnGpf.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { mDatabase.child("val").removeValue(); }
        });
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.gpfd:
                Intent intent=new Intent(MainActivity.this
                    ,MapsActivity.class);
                startActivity(intent);
                break;
        }
    }
}
```

Anexo 2.2 Código interfaz secundaria (Pantalla 2).

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    private DatabaseReference mDatabase;
    private ArrayList<Marker> tmpRealTimeMarkers = new ArrayList<>();
    private ArrayList<Marker> realTimeMarkers = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback this);
        mDatabase = FirebaseDatabase.getInstance().getReference();
    }
}
```

Anexo 2.3 Código de importación de Realtime Database de Firebase, filtrado de datos y aplicación de la fórmula de Haversine.

```

mDatabase.child("val").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for (Marker marker:realTimeMarkers){
            marker.remove();
        }
        for (DataSnapshot snapshot: dataSnapshot.getChildren()) {
            MapPojo mp= snapshot.getValue(MapPojo.class);
            Double latitud =mp.getLatitude();
            Double longitud = mp.getLongitude();
            Double r = 6378.10;
            Double dlat=latitud-(-0.3307);
            Double dlon=longitud-(-78.558327);
            Double a=(pow(sin((dlat)/2), 2))+(cos(-0.23052))*cos(latitud)*(pow(sin((dlon)/2), 2));
            Double d=2*r*asin(sqrt(a));
            MarkerOptions markerOptions=new MarkerOptions();

            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(latitud,longitud), 10));
            if (d<1.5){

                markerOptions.position(new LatLng(latitud,longitud)).icon(BitmapDescriptorFactory.fromVector(getApplicationContext().getDrawable(R.drawable.ic_baseline_widerg_24)));
                tpRealTimeMarkers.add(mMap.addMarker(markerOptions));
                Toast.makeText(getApplicationContext(), "En Tu camino de casa", Toast.LENGTH_LONG).show();
            } else{
                markerOptions.position(new LatLng(latitud,longitud)).icon(BitmapDescriptorFactory.fromVector(getApplicationContext().getDrawable(R.drawable.ic_baseline_widerg_24)));
                tpRealTimeMarkers.add(mMap.addMarker(markerOptions));
                Toast.makeText(getApplicationContext(), "En la falla", Toast.LENGTH_LONG).show();
            }
            /*Toast.makeText(getApplicationContext(), "Actualizando...", Toast.LENGTH_LONG).show();*/
        }

        realTimeMarkers.clear();
        realTimeMarkers.addAll(tpRealTimeMarkers);
    }
}

```

Anexo 2.4 Código personalización de marcadores y detección de errores.

```

private BitmapDescriptor bitmapDescriptorFromVector(Context context,int vectorResId){
    Drawable vectorDrawable= ContextCompat.getDrawable(context,vectorResId);
    vectorDrawable.setBounds( left: 0, top: 0,vectorDrawable.getIntrinsicWidth(),
        vectorDrawable.getIntrinsicHeight());
    Bitmap bitmap= Bitmap.createBitmap(vectorDrawable.getIntrinsicWidth(),
        vectorDrawable.getIntrinsicHeight(),Bitmap.Config.ARGB_8888);
    Canvas canvas=new Canvas(bitmap);
    vectorDrawable.draw(canvas);
    return BitmapDescriptorFactory.fromBitmap(bitmap);
}

@Override
public void onCancelled(@NonNull DatabaseError error) {

}

```


Anexo 2.5 Código botones, imágenes y textos de las interfaces.

Anexo 2.5.1 Botones Interfaz Principal (Pantalla 1).

```
<Button
    android:id="@+id/gpfd"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="492dp"
    android:text="Ir Al Mapa"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/gpf"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="0dp"
    android:text="Reset"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/gpfd" />
```

Anexo 2.5.2 Importación de imagen y texto interfaz principal.

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="178dp"
    android:layout_height="201dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.601"
    app:srcCompat="@mipmap/descarga" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Proyecto Técnico"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.146" />
```

Anexo 2.5.3 Texto interfaz principal parte 1.

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Jimmy Rojas"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.048"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.344" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Juan Diaz"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.045"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.393" />
```

Anexo 2.5.4 Texto interfaz principal parte 2.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Universidad Politécnica Salesiana"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.037" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Integrantes:"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.266" />
```

Anexo 2.5.5 Texto interfaz principal parte 3.

```
<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ingeniería Electrónica"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.096" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="GrandPa Finder"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.21" />
```